OAAS

Objects as a Service

$1T

Global business software and services market size by 2029

No progress since 1980s

Software development technology is still stuck with languages and approaches to build software developed in ‘80s.

More scalable horses, no automobiles.

Evolution →

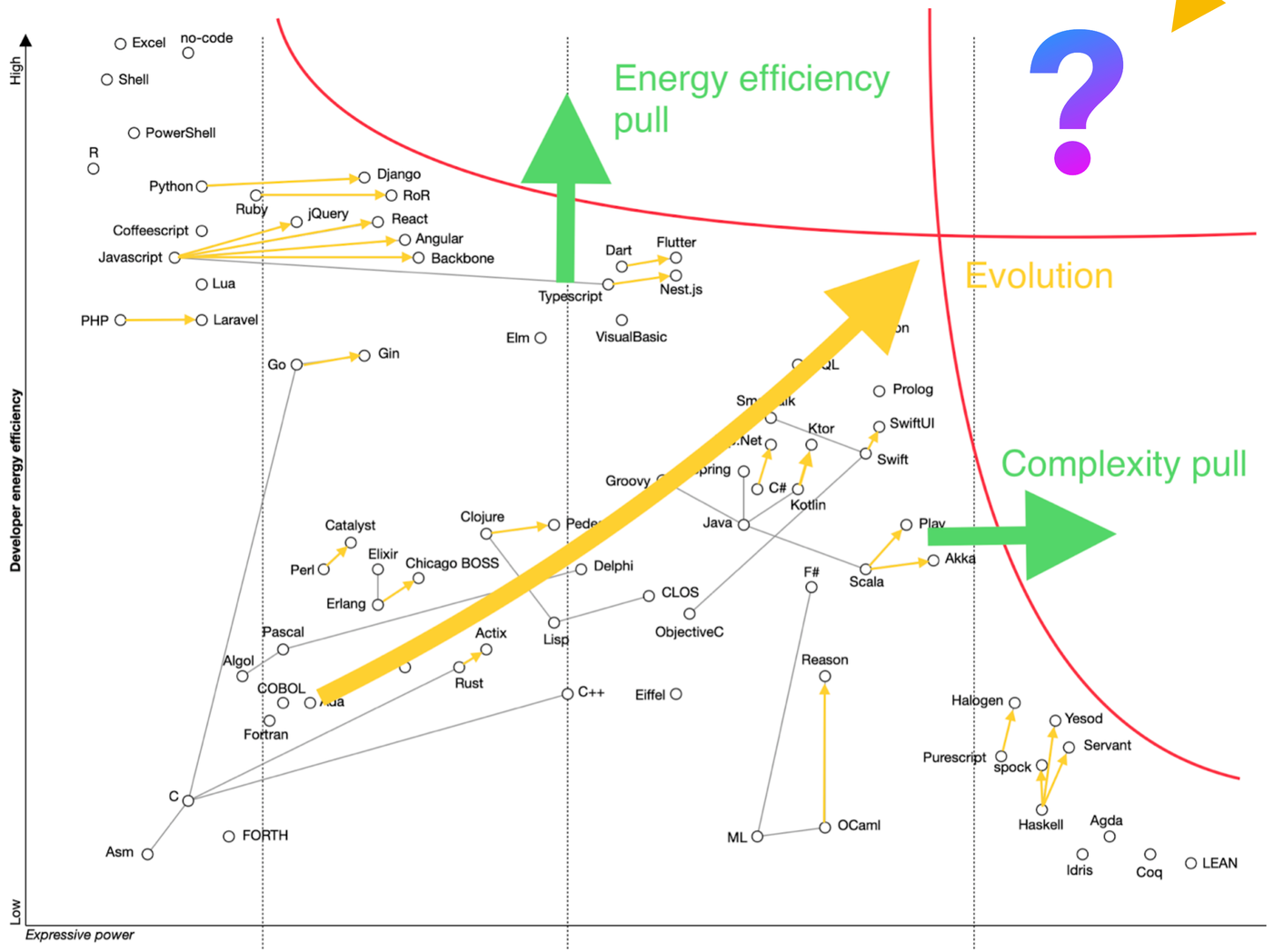
Need for expressive power and human energy efficiency are the main evolutionary forces pulling the industry to the future

←Future

“Instead of innovating out from the present, what you want to do is invent the future from the future. Go out and live in the future and bring the future back.”

— Alan Kay

AGI*



traversable.space/#essays/guts-d

* AGI will need better software development technologies too

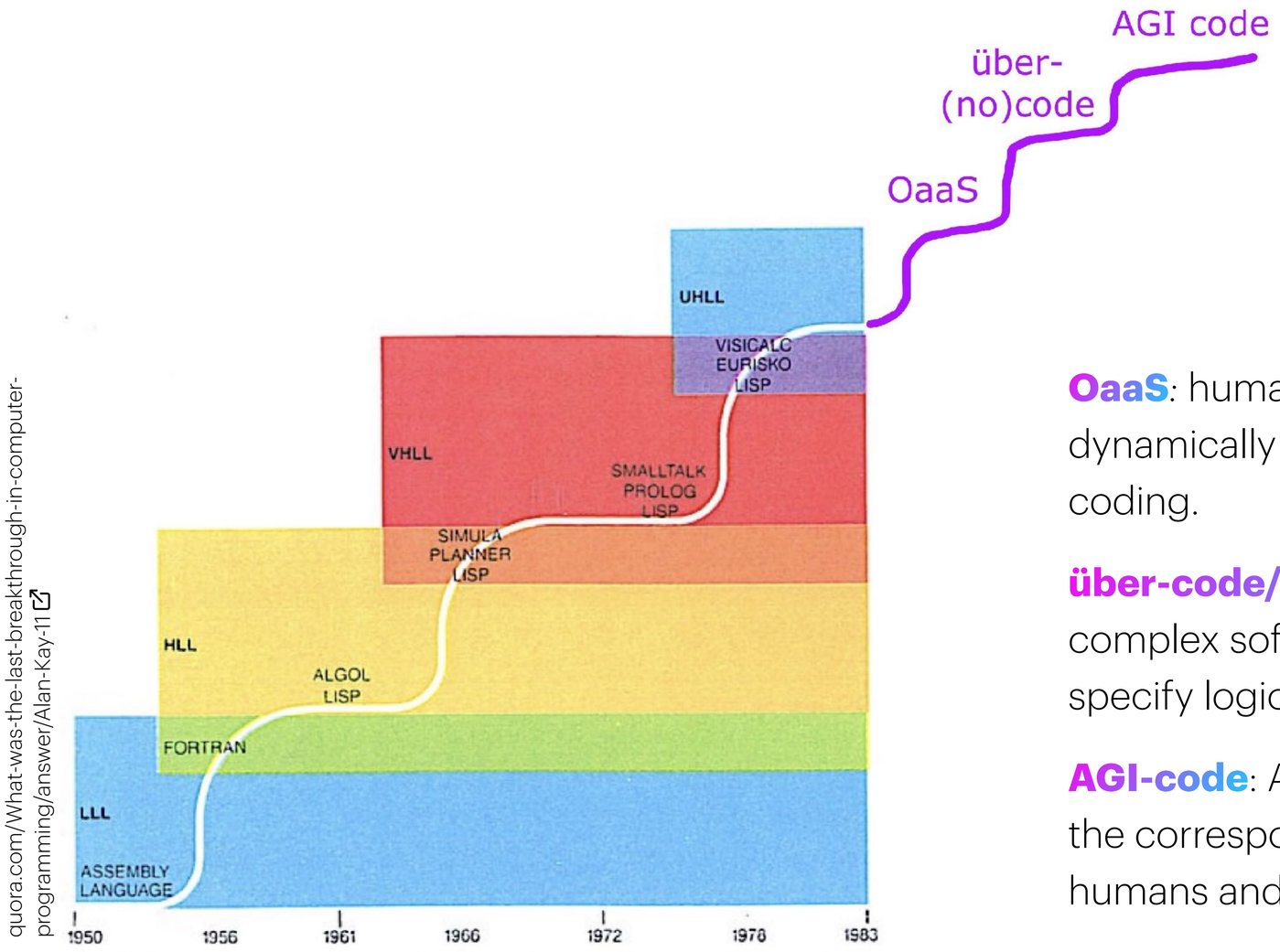
Complex Systems

Human Language

Build software using “organizing complex systems” approach and human language to define knowledge structure required to describe systems

OAAS → über-no/code → AGI-code

Build *systems* with no-code → build systems with *narratives* → build systems with *human language* → have *AGI* build systems



quora.com/What-was-the-last-breakthrough-in-computer-programming/answer/Alan-Kay-11

OaaS: humans build complex software *systems* out of dynamically created and composed objects, without coding.

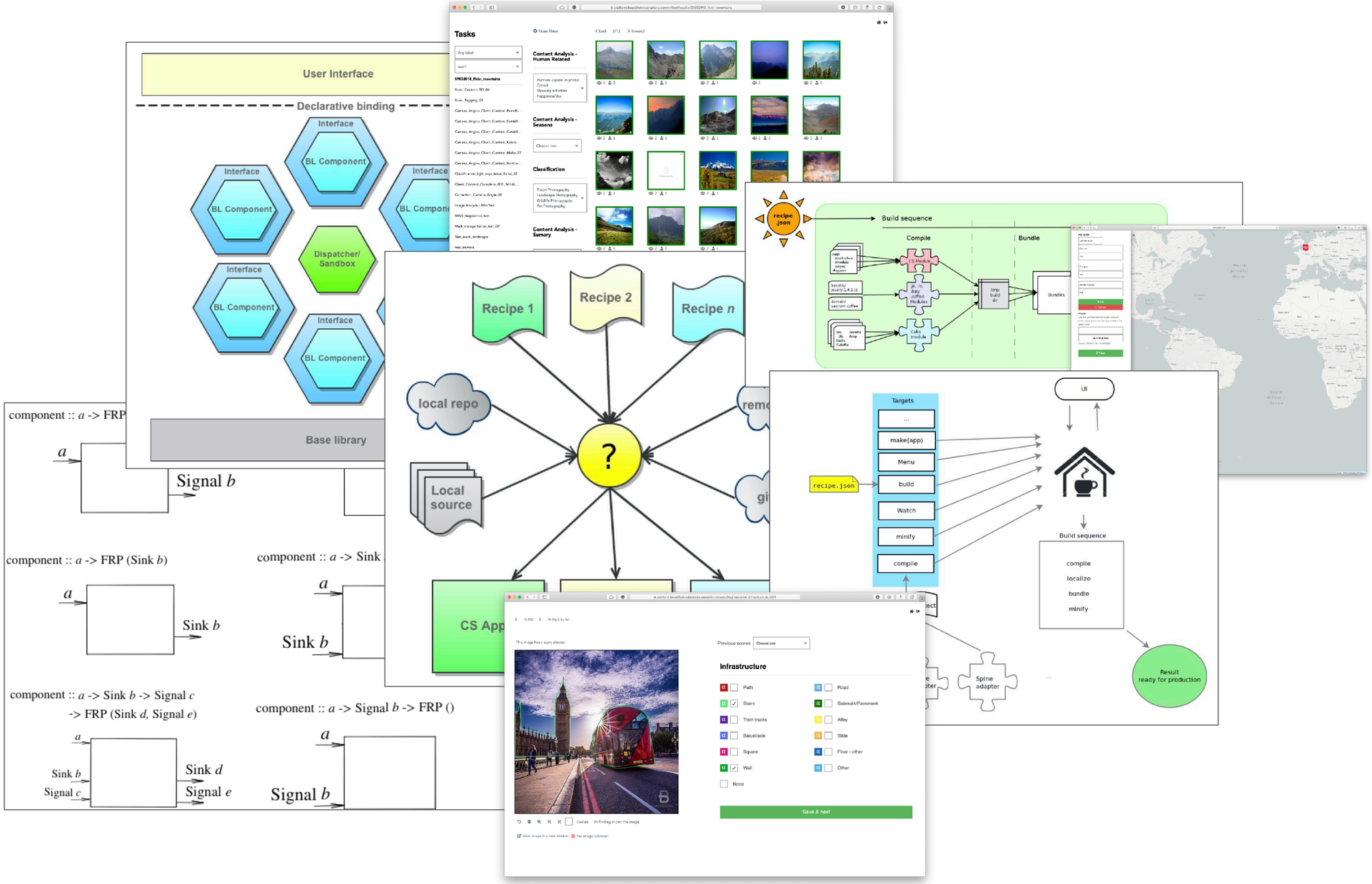
über-code/über-nocode: humans+*AI assistants* build complex software systems using human language to specify logical models, constraints and attractors.

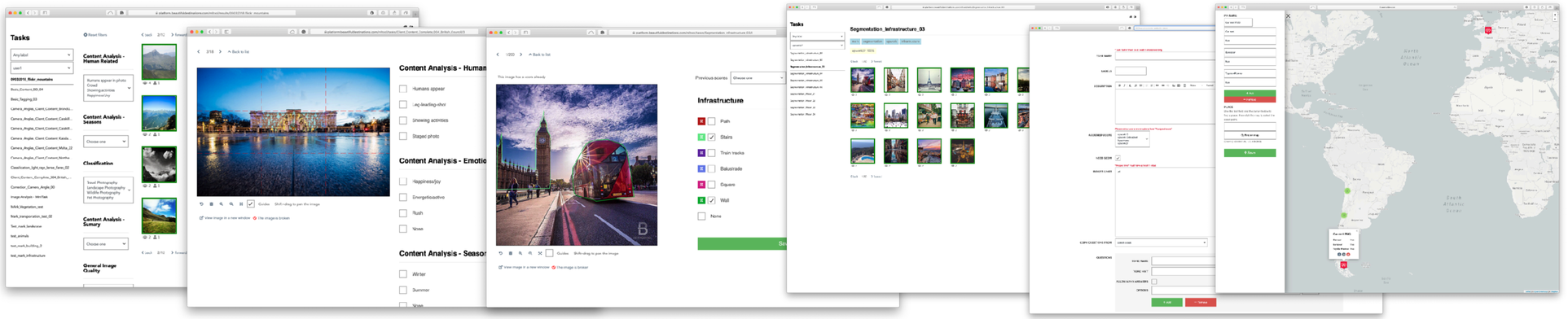
AGI-code: AGI builds and updates logical models and the corresponding software by interacting with humans and the environment.

Proven with prototypes

Did indeed enable building and evolving complex systems 10x faster,

by non-programmers


User Interface
Declarative binding
Interface
BL Component
Dispatcher/
Sandbox
Base library
component :: a -> FRP
a
Signal b
component :: a -> FRP (Sink b)
Sink b
component :: a -> Sink
component :: a -> Sink b -> Signal c
-> FRP (Sink d, Signal e)
Sink d
Signal e
component :: a -> Signal b -> FRP ()
Signal b
Tasks
Content Analysis - Human Related
Content Analysis - Seasons
Classification
Content Analysis - Summary
Recipe 1
Recipe 2
Recipe n
local repo
Local source
?
CS App
recipe.json
Build sequence
Compile
Bundle
Bundles
Targets
make(app)
Menu
build
Watch
minify
compile
UI
Build sequence
compile
localize
bundle
minify
Result ready for production
Spine adapter
Infrastructure
Path
Road
Stairs
Sidewalk/Pavement
Train tracks
Alley
Balustrade
Slide
Square
Floor - other
Wall
Other
None
Save & next



Built in 3 days

An application which is more powerful and feature-rich than the main product of a startup with $20M funding after 2 years of development. Used in production in 3 days instead of a $18k contract with the aforementioned company.

New industry

Everyone able to step-by-step define a logical domain model is able to build software.

“Instead of innovating out from the present, what you want to do is invent the future from the future. Go out and live in the future and bring the future back.”

“Computer science inverts the normal. In normal science, you're given a world, and your job is to find out the rules. In computer science, you give the computer the rules, and it creates the world.”

“If you want to make money, don’t make a startup, start an industry.”

— Alan Kay

EUGENE NAUMENKO

eugene@traversable.space

https://traversable.space